

Structured Adaptive Mesh Refinement and Multilevel Preconditioning for Non-Equilibrium Radiation Diffusion

Bobby Philip
T-7, Mathematical Modeling and Analysis
Theoretical Division, Los Alamos National Laboratory

April 10, 2008

Non-Equilibrium Radiation-Diffusion

Preconditioned Newton-Krylov Methods

Structured Adaptive Mesh Refinement

Numerical Results

Non-Equilibrium Radiation-Diffusion

Model equations:

$$\frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) = \sigma_a (T^4 - E) \quad \text{in } \Omega = [0, 1]^d$$

$$\frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) = -\sigma_a (T^4 - E) \quad \text{in } \Omega = [0, 1]^d$$

Non-Equilibrium Radiation-Diffusion

Model equations:

$$\frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) = \sigma_a (T^4 - E) \quad \text{in } \Omega = [0, 1]^d$$

$$\frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) = -\sigma_a (T^4 - E) \quad \text{in } \Omega = [0, 1]^d$$

Constitutive law:

$$\sigma_a = \frac{z^3}{T^3}$$

Non-Equilibrium Radiation-Diffusion

Model equations:

$$\frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) = \sigma_a (T^4 - E) \quad \text{in } \Omega = [0, 1]^d$$

$$\frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) = -\sigma_a (T^4 - E) \quad \text{in } \Omega = [0, 1]^d$$

Constitutive law:

$$\sigma_a = \frac{z^3}{T^3}$$

Diffusion coefficients:

$$D_r = \frac{1}{\left(3\sigma_a + \frac{\|\nabla E\|}{E}\right)}$$

$$D_t = kT^{5/2}$$

Non-Equilibrium Radiation-Diffusion

Model equations:

$$\frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) = \sigma_a (T^4 - E) \quad \text{in } \Omega = [0, 1]^d$$

$$\frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) = -\sigma_a (T^4 - E) \quad \text{in } \Omega = [0, 1]^d$$

Non-Equilibrium Radiation-Diffusion

Model equations:

$$\frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) = \sigma_a (T^4 - E) \quad \text{in } \Omega = [0, 1]^d$$

$$\frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) = -\sigma_a (T^4 - E) \quad \text{in } \Omega = [0, 1]^d$$

Initial conditions:

$$E = E_0, \quad T = (E_0)^{1/4} \quad \text{at } t = 0$$

Non-Equilibrium Radiation-Diffusion

Model equations:

$$\frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) = \sigma_a (T^4 - E) \quad \text{in } \Omega = [0, 1]^d$$

$$\frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) = -\sigma_a (T^4 - E) \quad \text{in } \Omega = [0, 1]^d$$

Initial conditions:

$$E = E_0, \quad T = (E_0)^{1/4} \quad \text{at } t = 0$$

Boundary conditions:

$$\frac{1}{2} \mathbf{n} \cdot D_r \nabla E + \frac{E}{4} = R \quad \text{on } \partial\Omega_{\mathcal{R}}, \quad t \geq 0$$

$$\mathbf{n} \cdot D_r \nabla E = 0 \quad \text{on } \partial\Omega_{\mathcal{N}}, \quad t \geq 0$$

$$\mathbf{n} \cdot \nabla T = 0 \quad \text{on } \partial\Omega, \quad t \geq 0$$

Previous Work

- ▶ Rider, Knoll and Olson (JQSRT, **63**, 1999; JCP, **152**, 1999) introduced the idea of physics based preconditioning in 1D
- ▶ Mousseau, Knoll, Rider (JCP, 2000) and Mousseau, Knoll (JCP, 2003) demonstrated effectiveness for 2D problems
- ▶ Mavriplis (JCP, **175**, 2002) compared Newton-Multigrid and FAS using agglomeration ideas on unstructured grids.
- ▶ Stals (ETNA, **15**, 2003), Newton-Multigrid and FAS, local refinement on *unstructured* grids for equilibrium radiation diffusion.
- ▶ Lowrie (JCP, 2004) compares different time integration methods for non-equilibrium radiation diffusion

Previous Work

- ▶ Brown, Shumaker, Woodward (JCP, 2005) consider fully implicit methods and high order time integration.
- ▶ Shestakov, Greenough, and Howell (JQSRT, 2005) consider pseudo-transient continuation on AMR grids using an alternative formulation.
- ▶ Glowinski, Toivanen (JCP, 2005) consider using automatic differentiation and system multigrid.
- ▶ Pernice, Philip (SISC, 2006), use JFNK with FAC preconditioners on AMR grids for equilibrium radiation-diffusion on SAMR grids.

Discretization

- ▶ Time Discretization : BDF2
 - ▶ constant fixed timestep
 - ▶ constant final timestep
 - ▶ limiting relative change
 - ▶ estimating dynamical scale (Rider-Knoll approach)

Discretization

- ▶ Time Discretization : BDF2
 - ▶ constant fixed timestep
 - ▶ constant final timestep
 - ▶ limiting relative change
 - ▶ estimating dynamical scale (Rider-Knoll approach)
- ▶ Space Discretization:
 - ▶ Cell Centered Finite Volume Discretization
 - ▶ Cell centered diffusion coefficients averaged to cell faces
 - ▶ Fluxes are computed at cell faces
 - ▶ Discontinuities are aligned with cell faces

Inexact Newton Methods

- ▶ Let $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and consider solving $F(x) = 0$.

Inexact Newton Methods

- ▶ Let $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and consider solving $F(x) = 0$.
- ▶ The k^{th} step of classical Newton's method requires solution of the Newton equations:

$$F'(x_k)s_k = -F(x_k).$$

Inexact Newton Methods

- ▶ Let $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and consider solving $F(x) = 0$.
- ▶ The k^{th} step of classical Newton's method requires solution of the Newton equations:

$$F'(x_k)s_k = -F(x_k).$$

- ▶ With *inexact Newton methods*, we only require

$$\|F(x_k) + F'(x_k)s_k\| \leq \eta_k \|F(x_k)\|, \quad \eta_k > 0.$$

This can be done with any iterative method.

Iterative Linear Solvers

- ▶ System MG could be used directly, or
- ▶ Krylov subspace methods - need Jacobian-vector products, which can be approximated by

$$F'(x_k)v \approx \frac{F(x_k + \varepsilon v) - F(x_k)}{\varepsilon}, \quad \varepsilon \approx \mathcal{O}(\sqrt{\epsilon_{\text{mach}}}).$$

- ▶ The resulting *Jacobian-free Newton-Krylov* (JFNK) method is easy to use because only function evaluation and preconditioning setup/apply are needed.

Preconditioned Krylov Methods

- ▶ Right-preconditioning of the Newton equations is used, i.e., we solve

$$(J(x_k)P^{-1})Ps_k = -F(x_k).$$

where P is the preconditioner.

Preconditioned Krylov Methods

- ▶ Right-preconditioning of the Newton equations is used, i.e., we solve

$$(J(x_k)P^{-1})Ps_k = -F(x_k).$$

where P is the preconditioner.

- ▶ For JFNK this requires the Jacobian-vector products:

$$(J(x_k)P^{-1})v \approx \frac{F(x_k + \varepsilon P^{-1}v) - F(x_k)}{\varepsilon}.$$

Preconditioned Krylov Methods

- ▶ Right-preconditioning of the Newton equations is used, i.e., we solve

$$(J(x_k)P^{-1})Ps_k = -F(x_k).$$

where P is the preconditioner.

- ▶ For JFNK this requires the Jacobian-vector products:

$$(J(x_k)P^{-1})v \approx \frac{F(x_k + \varepsilon P^{-1}v) - F(x_k)}{\varepsilon}.$$

- ▶ The approximate Jacobian-vector is computed in two steps:
 - ▶ Solve $y = P^{-1}v$ approximately
 - ▶ Compute $\frac{F(x_k + \varepsilon y) - F(x_k)}{\varepsilon}$.

Linear Systems

JFNK allows us to focus on developing effective preconditioners.
The Jacobian systems at each Newton step are of the form:

$$\mathcal{L} \begin{pmatrix} \delta E \\ \delta T \end{pmatrix} = \begin{pmatrix} -r_E \\ -r_T \end{pmatrix}$$

where

$$\mathcal{L} = \begin{pmatrix} \frac{I}{\Delta t} - \nabla \cdot D_r^k \nabla + \sigma_a I & -\sigma_a (T^k)^3 \\ -\sigma_a I & \frac{I}{\Delta t} - \nabla \cdot D_t^k \nabla + \sigma_a (T^k)^3 \end{pmatrix}$$

Operator Split Preconditioner

We use a splitting of the form shown in our preconditioner

$$\mathcal{L} \approx \mathcal{P}_1 \mathcal{P}_2$$

where

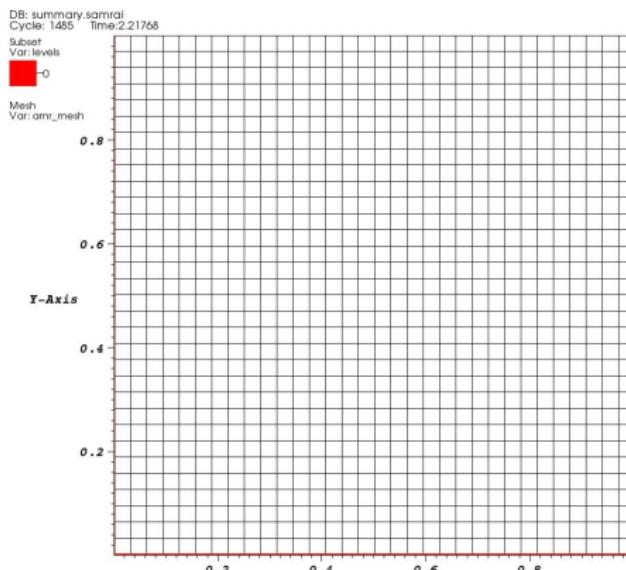
$$\mathcal{P}_1 = \begin{pmatrix} \frac{I}{\Delta t} - \nabla \cdot D_r^k \nabla & 0 \\ 0 & \frac{I}{\Delta t} - \nabla \cdot D_t^k \nabla \end{pmatrix}$$

and

$$\mathcal{P}_2 = \begin{pmatrix} (1 + \Delta t \sigma_a)I & -\Delta t \sigma_a (T^k)^3 \\ -\Delta t \sigma_a I & I + \Delta t \sigma_a (T^k)^3 \end{pmatrix}$$

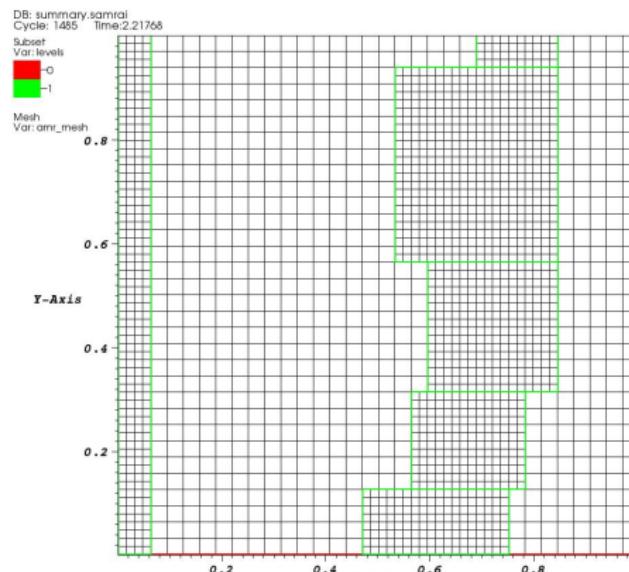
Structured Adaptive Mesh Refinement

Structured adaptive mesh refinement (SAMR) represents a locally refined mesh as a union of logically rectangular meshes.



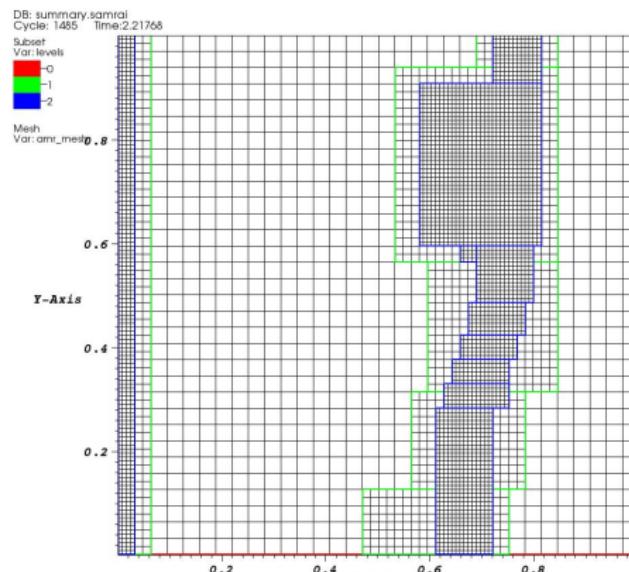
Structured Adaptive Mesh Refinement

Structured adaptive mesh refinement (SAMR) represents a locally refined mesh as a union of logically rectangular meshes.



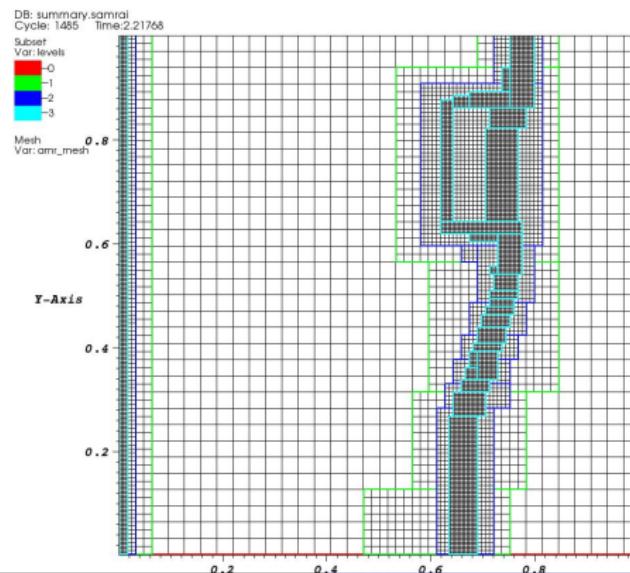
Structured Adaptive Mesh Refinement

Structured adaptive mesh refinement (SAMR) represents a locally refined mesh as a union of logically rectangular meshes.



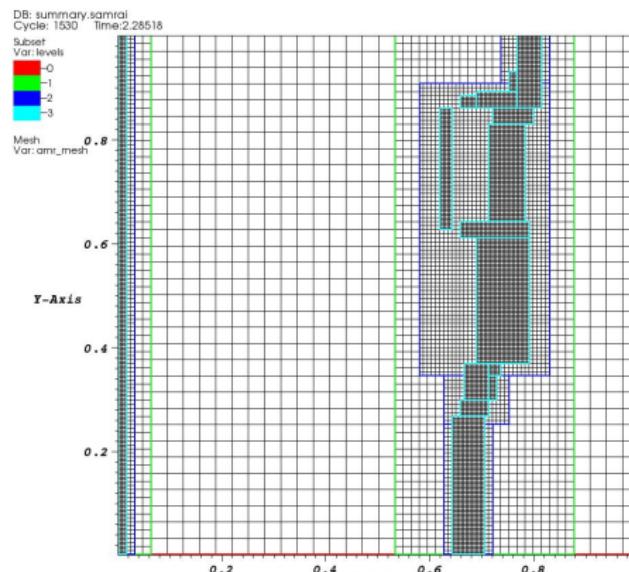
Structured Adaptive Mesh Refinement

Structured adaptive mesh refinement (SAMR) represents a locally refined mesh as a union of logically rectangular meshes.



Structured Adaptive Mesh Refinement

Structured adaptive mesh refinement (SAMR) represents a locally refined mesh as a union of logically rectangular meshes.



Simulation Software

- ▶ SAMRAI package for AMR
- ▶ PETSc SNES package for inexact Newton
- ▶ PETSc Krylov solver - GMRES
- ▶ ML solvers package for multilevel preconditioners and operators - FAC, AFACx, MDS

Simulation parameters

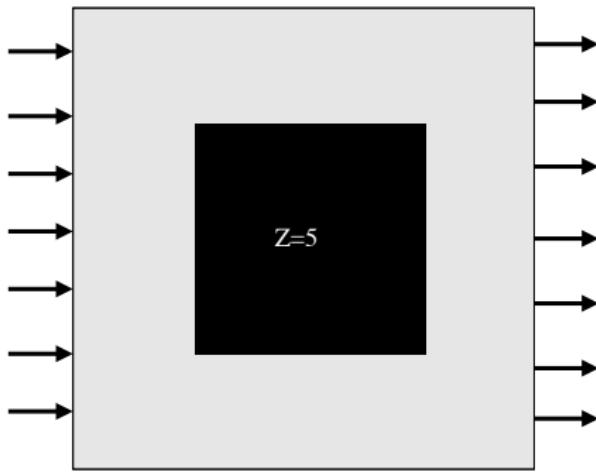
► Solver parameters

- absolute tolerance: $1.0e - 07$
- relative tolerance: $1.0e - 07$
- step tolerance: $1.0e - 10$
- max. gmres subspace dimension: 20
- max. linear iterations: 25
- forcing term: $\eta_k = 0.1$
- final time: 5.0

► SAMR parameters

- refinement ratio: 2
- combine efficiency: 0.85
- error tagging: based on curvature and gradient of E

Case 1



Iteration counts for Case 1

Table: Number of linear iterations, $k=0.1$

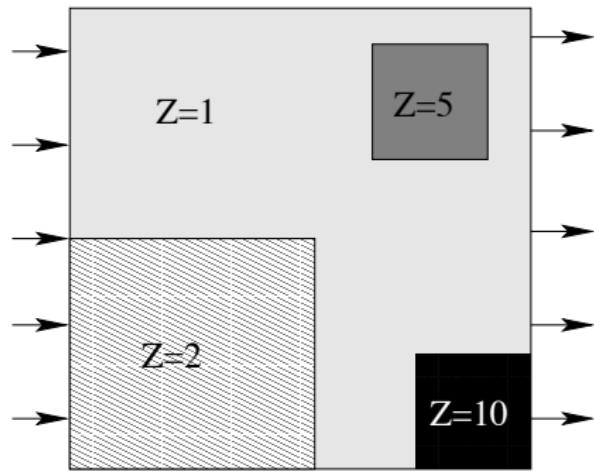
Levels	1	2	3	4	5
16×16	13.9	12.6	12.5	11.8	11.9
32×32	12.5	12.3	10.5	10.2	-
64×64	12.1	9.8	9.3	-	-
128×128	9.2	9.1	-	-	-
256×256	8.0	-	-	-	-

Iteration counts for Case 1

Table: Number of nonlinear iterations, $k=0.1$

Levels	1	2	3	4	5
16×16	3.4	3.0	2.8	2.7	3.1
32×32	3.0	2.8	2.4	2.6	-
64×64	2.8	2.3	2.3	-	-
128×128	2.2	2.2	-	-	-
256×256	2.0	-	-	-	-

Case 2



Case 2

Iteration counts for Case 2

Table: Number of linear iterations, Case 2, k=0.1

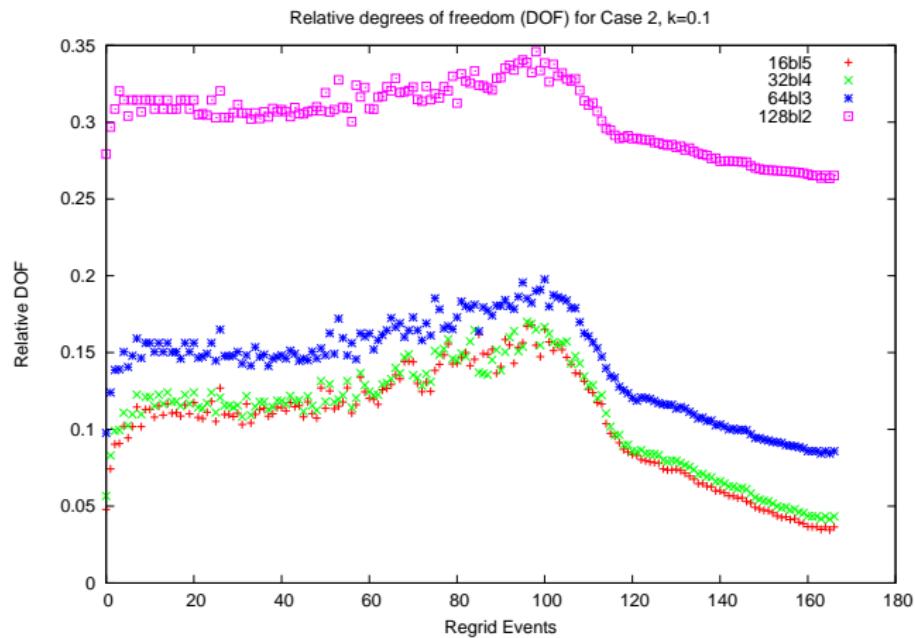
Levels	1	2	3	4	5
16×16	27.2	26.1	18.8	12.6	10.3
32×32	25.3	18.1	11.5	8.5	-
64×64	16.4	10.7	7.9	-	-
128×128	9.6	7.5	-	-	-
256×256	6.5	-	-	-	-

Iteration counts for Case 2

Table: Number of nonlinear iterations, Case 2, k=0.1

Levels	1	2	3	4	5
16×16	3.4	3.3	2.9	2.8	2.7
32×32	3.3	2.8	2.5	2.3	-
64×64	2.8	2.4	2.0	-	-
128×128	2.3	1.9	-	-	-
256×256	1.8	-	-	-	-

Comparison of DOF's for Case 2



Wall clock comparisons

Table: Wall clock timings in seconds

Problem	Total Time	Preconditioner	Function Evaluation
32b4l	2124	968 (45%)	537 (25%)
64b3l	2501	1088 (43%)	531 (21%)
128b2l	4633	2053 (44%)	736 (15%)
256b1l	12288	5536 (45%)	1326 (10%)

Additive preconditioning

Table: AFACx Preconditioner: Linear iterations

Levels	1	2	3	4	5
16×16	16.7	13.7	7.3	5.9	6.8
32×32	11.7	6.9	4.4	5.3	-
64×64	6.0	4.0	3.6	-	-
128×128	3.2	3.2	-	-	-
256×256	2.3	-	-	-	-

Additive preconditioning

Table: MDS Preconditioner: Linear iterations

Levels	1	2	3	4	5
16×16	16.7	15.6	12.0	8.3	9.4
32×32	11.7	10.5	7.5	7.8	-
64×64	6.0	6.2	6.7	-	-
128×128	3.2	5.8	-	-	-
256×256	2.3	-	-	-	-

Performance comparison

Table: Comparison of linear iteration counts with preconditioners

Grids	256b1l	128b2l	64b3l	32b4l	16b5l
FAC	2.3	2.6	2.7	2.8	3.0
AFACx	2.3	3.2	3.6	5.3	6.8
MDS	2.3	5.8	6.7	7.8	9.4

Conclusions and Future Work

- ▶ Conclusions
 - ▶ Developed an efficient solver for non-equilibrium radiation diffusion on AMR grids
 - ▶ Performance of multiplicative and additive preconditioners
- ▶ Future work/Possible improvements
 - ▶ Better discretizations for AMR grids for problems with discontinuous coefficients
 - ▶ Improved performance of preconditioners
 - ▶ Error estimation for finite volume discretizations
 - ▶ AMR Grid alignment